● PRINTER RUSH ●
(PTO ASSISTANCE)

Application : 10684614 **Examiner :** Nguyen **GAU :** 2182

From: NB **Location:** [✓ IDC] FMF FDC **Date:** 09/21/05

Tracking #: opm10684614 Week Date: 08/08/05

DOC CODE	DOC DATE	MISCELLANEOUS
☐ 1449		☐ Continuing Data
☐ IDS		☐ Foreign Priority
☐ CLM		☐ Document Legibility
☐ IIFW		☐ Fees
☐ SRFW		☐ Other
☐ DRW		
☐ OATH		
☐ 312		
☑ SPEC	10/14/03	

[RUSH] **MESSAGE:** Please provide missing Serial NO. and date on the following pages:

a) page 23, lines 1 and 2 10684871 10/14/03
b) page 37, line 10 10685137 " "
c) page 38, line 4. 10684916, " "

Thankyou

[XRUSH] **RESPONSE:**

Done

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

Route Hardware With Parallel Routing Scheme" by L. Moll, Ser. No. 10/684871, filed 10/14/03, and assigned to Broadcom Corporation, which is also the assignee of the present application, and is hereby incorporated by reference in its entirety.

[076] Turning now to Figure 4, a block diagram illustrating one embodiment of virtual channels in the system 300 is shown, as well as examples of splitting and merging packet traffic. In the illustrated embodiment, the receive interface circuits 330-332 and the transmit circuits 350-352 are shown. Additionally, the packet manager circuit 320 is shown as including an input packet manager circuit (PMI) 322 and an output packet manager circuit (PMO) 324. The PMI 322 is coupled to transmit write commands on the bus 130 to write received packets to memory. The PMO 324 is coupled to transmit read commands on the bus 130 and to receive the read data comprising packets to be transmitted from the system 300. Additionally, as mentioned above, each of the PMI 322 and the PMO 324 may be configured to read and write descriptors defining the locations in memory to which the packets are to be read and written. Upon completion of data transfer specified by a descriptor, each descriptor is updated and released to software by writing the descriptor back to memory. For purposes of providing efficient descriptor write back operations, the descriptors may be collected until a predetermined number of descriptors are accumulated or a descriptor timer 375 expires, whichever occurs first. Each of these events (packet transfer, descriptor processing, errors) and others may cause interrupts to be issued by the packet manager 320 which are issued after a predetermined number of packets are processed or an interrupt timer 375 expires, whichever occurs first.

[077] Each receive circuit 330-332 supports a set of input virtual channels (IVCs) defined by the interface from which the receive circuit receives packet data. For example, the SPI-4 interface and the HT interface may both support 16 virtual channels in hardware (although more may be used by software in the SPI-4 interface, since an 8-bit virtual channel value is supported). Thus, each receive circuit 330-332 supports 16 IVCs (numbered 0-15 in Figure 4). Similarly, each transmit circuit 350-352 supports 16 output virtual channels (OVCs), numbered 0-15 in Figure 4. Other embodiments may employ more or fewer IVCs and OVCs according to the interfaces supported by those embodiments.

[078] The PMI 322 includes a logical set of input queues (e.g. 32 in the illustrated embodiment, numbered 0-31, although more or fewer input queues may be included in other

owner). To take advantage of this property of write-invalidate commands, the packet manager input 540 will release descriptors by increments of two to reduce descriptor bandwidth. To promote full cache line transfers, timer 530 is set after the first descriptor in a cache line is ready to be released to the software. If the other descriptor in the same cache line is finished before the timer 530 expires, both descriptors will be released together with a write-invalidate command on the bus 130. If the timer 530 expires, then both descriptors will be written back one by one with read-modify-write commands (read-exclusive followed by write) when they are ready to be released. The descriptor write back mechanism can be implemented by the structures disclosed in copending U.S. patent application entitled "Descriptor Write Back Delay Mechanism To Improve Performance" by K. Oner, Ser. No.10/685137, filed 10/14/03, and assigned to Broadcom Corporation, which is also the assignee of the present application, and is hereby incorporated by reference in its entirety.

[0118] While multiple individual counter circuits could be used to implement the interrupt and descriptor timers for multiple channels (e.g., 64 virtual channels), such a solution would consume valuable chip area and would increase the system complexity, especially where independent and different time-out settings are required for multiple channels. Accordingly, an exponential channelized timer is advantageously used in connection with multi-channel, multiprocessor applications such as depicted in Figures 3-5 to efficiently provide a programmable timer with individual time-out settings for multiple channels. In a selected embodiment, an exponential channelized timer monitors a selected bit position of a free-running timer and generates a pulse whenever a transition is observed at that bit location. In this embodiment, the time-out values that can be set are exponential values (power of 2), so the exponential channelized timer acts as an interval timer where the timer accuracy goes down as the interval increases. For example, if an exponential channelized timer for a particular channel is programmed to monitor bit location number five of a free running 32-bit counter, then the time-out will be generated within a time interval of 32 (2^5) and 63 (2^6-1), reflecting the fact that the free running timer is not reset with each packet. As will be appreciated, this interval increases exponentially as the monitored bit location register becomes more significant. Therefore, the timer gets less accurate as this value is increased. However, by using multiplexers and control registers for each channel, the exponential timer can be used to generate time-outs for multiple channels without requiring long timer counters for each channel, using only a single free running counter. The timer module 375 can be implemented by the structures disclosed in copending U.S. patent application entitled "Exponential Channelized Timer" by K. Oner, Ser. No.10/684916, filed 10/14/03, and assigned to Broadcom Corporation, which is also the assignee of the present application, and is hereby incorporated by reference in its entirety.

[0119] As described above, there are at least two ways for the software to find out that new packets have arrived, including spinning on the descriptors and checking whether their hardware bits are reset, and waiting for the packet manager to interrupt the processor. In addition, the processor can read the descriptor control register to check on the status of the data transfer. When descriptor spinning is used, the software will spin on the descriptors that it has released to the packet manager 320. When descriptors are loaded as shared by the spinning processors, no additional bus traffic is generated on the bus 130 after the initial read. In particular, the packet manager 320 may release a descriptor by writing the descriptor to the memory with either a write invalidate or read-exclusive and write command, at which time the spinning processor's cached copy will be invalidated, and only then it will fetch the updated descriptor. The software can check the hardware bit 702 to determine if this descriptor is done or not. In a selected embodiment, the descriptor is done if the hardware bit 702 is reset. As a result, the work of processing the received packets can be dispatched to each processor 342-345 in groups of descriptors. By having the packet manager 320 release all of the descriptors in order, each processor can spin on the last descriptor it owns, and when the software detects that the last descriptor it owns is done, it can assume that all the previous descriptors are done as well.

[0120] In accordance with the present invention, the descriptor structure has additional descriptor fields specifying descriptor ownership and whether the descriptor is for the start or end of the packet. With these fields, the first descriptor can separately specify descriptor ownership (with the HW bit 702) and whether the descriptor is for the start of packet (with the SOP bit 704). In addition, an end-of-packet indicator (EOP bit 706) is used to specify when the descriptor's buffer contains the end of the packet. As a result, a whole packet can be specified by a single descriptor (in which case both the SOP and EOP bits are set, such as with descriptor 851) or by multiple descriptors (in which case the first descriptor, e.g., 852, has its SOP bit set,